AUCTIONS FOR MULTIPLE ITEMS WITH
CONSTRAINTS SPECIFIED BY THE BIDDERS

Field of the Invention

We disclose a method for considering constraints imposed by the participants in auctions for

5   multiple items. We include representative examples of such constraints. We demonstrate that the

winner determination problem may be formulated as an integer program, and may be solved by

commercially available software packages. This invention includes both a business process of

auctioning multiple items with constraints specified by the bidder, and a computer implemented

method for determining winners in such an auction.

Summary of the Invention

10   Summary of the Invention

A Problem Solved by the Invention

Auctions have been used in the sale of items for many years. Auctions provide a means of making

scare goods available to a large audience, and of ensuring that the seller receives the maximum

revenue for the goods. Many forms of auctions exist, including open-cry auctions, silent auctions,

15   ascending bid auctions, and descending bid auctions. With the advent of the Internet and other

forms of electronic communications, auctions are becoming increasingly common.

Most auctions are for single items, or for multiple copies of the same item. Auctions for related items are often held simultaneously. E-bay.com and Amazon.com provide an auctioning capability through which individuals or companies can offer goods for auction. It is common for companies like eBay.com and Amazon.com to have many similar or related items on auction at the same

5   time. Examples of similar items include multiple copies of the same book, toy, or CD being offered for sale in different, simultaneously active, auctions by different sellers. Examples of related items include separate, but simultaneous, auctions (often by different sellers) of a table and matching chairs.

There are numerous instances where the value (to a bidder) of a set of goods may not equal the sum of the value (to that bidder) of the individual goods. In some cases, the value of a collection

10   of items, such as an airline ticket, a hotel room, and theater tickets, may be higher than the sum of the value of the individual items. In this case, the items (airline ticket, hotel, theater tickets) are said to complement one another. In other cases, the sum of the value of the individual items, such as theater tickets and concert tickets for the same time and date, may be higher than the value of

15   the combination. In this case, the items are said to substitute for one another. Therefore, a bidder, attempting to obtain a collection of items, or to obtain one collection from a set of specified collections, through participation in several single item auctions, is faced with a dilemma. She may bid on all the items in her desired collection, and obtain some, but not all, of the items in that collection. Even if the total amount she bids does not exceed her value for the collection, she may

20   end up paying more for the items she receives than she values that subset of the collection. Similarly, if she bids for both theater tickets and concert tickets, she may end up with both sets of tickets, at a cost greater than her value for the pair.

Finally, in small business/consumer to small business consumer auction such as eBay.com and Amazon.com. bidders often have a limited budget, or a limited spending target for certain classes of items. When a bidder over spends this budget, through unexpectedly winning more auctions than anticipated, he may not have sufficient funds to complete all the contracted transactions. This typically results in the bidder defaulting on some transactions. In some auctions, old "losing" bids get reinstated (potentially resulting in a cascade of defaults); in others; the item is simply re-listed for auction.

Seller side constraints may be appropriate in instances in which a seller wishes to sell multiple items, is indifferent as to whether items are sold to a single or multiple buyers, but wishes to sell all or most of the collection so as not to be left with only 1n value items.

Therefore, it is evident that in auction systems offering multiple items, there is a need for methods through which participants can specify constraints that describe or characterize the combination of items they desire to win or sell, as well as a need for methods to solve the winner determination problem in auction systems that permit participants to specify such constraints.

Accordingly, in a first aspect of the present invention, we now disclose a method comprising the steps of:

(1) establishing an auction system; and

(2) enabling the auction system so that it is responsive to constraints specified by or on behalf of a participant in the auction.

In a second aspect of the present invention, we disclose a program medium executable in a computer system for facilitating an auction, the program medium comprising steps for:

5   (1) establishing an auction system;

and

(2) enabling the auction system so that it is responsive to constraints specified by or on behalf of a participant in the auction.

Brief Description of the Drawing

10   The invention is illustrated in the accompanying drawing, in which:

FIG. 1 shows a system block diagram of a total auction system;

FIGs. 2a, 2b show a depiction of a user interface;

FIGs. 3a, 3b provide an illustration of an integer programming formulation;

FIG. 4 shows the Fig. 1 system where the winners may be determined by solving an initial

15   integer programming formulation;

FIG. 5 shows the Fig. 1 system where the winners may be determined by solving a column generation based integer programming formulation;

and

FIG. 6 shows an additional (optional) user interface that permits a bidder to review and

20   modify proposals.

Detailed Description of the Invention

We describe methods that allow a bidder in an auction system to specify constraints that describe or characterize the combinations of items he wishes to win. We demonstrate how these constraints can be used, together with item availability information and auction system objectives, to formulate an integer program that represents the winner determination problem.

We assume that multiple items are being offered for auction, and that the auction system itself provides a means for bidders to search through the items and to place bids on individual items. We also assume that the auction system provides a means for informing bidders of auctions they have won.

Each bidder can bid an arbitrary number of bids. Each bid is for a unique item and specifies the amount that the bidder is willing to pay for that item. Each bidder can also specify a set of constraints.

The simplest, but possibly most valuable constraint is a *budget constraint* that specifies the total amount the bidder is willing to pay for a collection of items on which he has placed bids. This constraint is relevant only if the total amount bid on the items in the collection is greater that the specified budget.

A *maximum quantity constraint* provides a generalization of the concept of substitutability. A bidder may place bids on $n \geq 2$ items, and express a willingness to win only $m < n$ of the items. We expect that this function would typically be used primarily for the case of similar items.

A *minimum quantity constraint* is used to represent factors such as economies of scale by the bidder. A bidder places bids on $n \geq 2$ items, and is willing to win any of these items if and only if he wins at least $m \leq n$ of them.

A *precedence constraint* is used to indicate that a bidder is willing to win one item only if he also wins another item. This can be generalized to a precedence set for an item: a bidder is willing to win an item, only if he also wins all of the $n \geq 1$ items in a precedent set specified for that item. We further generalize to *alternate precedence sets* in which a bidder is willing to win an item only if he wins all of the items in at least one of the $m \geq 1$ precedence sets specified for that item. A *general linear constraint* allows the bidder to reflect other considerations, such as total volume or weight, of items. The bidder specifies a coefficient (typically non-negative) a_i for an item i and maximum and/or minimum quantities, denoted c^U and c^L respectively, that indicate upper and lower bounds on the sum of the coefficients that of combinations of items that the bidder is willing to accept. More than one constraint of this form is allowed.

All of these constraints can be represented by linear relationships between indicator variables on the bids, with the inclusion of some auxiliary indicator variables, as will be demonstrated below. Other, non-linear constraints may also be of interest and are included in this invention. However, we note that for simplicity of use, and for solvability of the winner determination problem, a restriction to linear constraints is generally a desirable feature for an auction system.

Sellers may also express constraints. Typical seller constraints would include minimum revenue for a set of items (analogous to a "reserve price - but in a set), minimum quantity to be sold in the

case of a collection of similar or identical items, and precedence constraints, stating that one item should be sold only if another item is also sold.

For completeness, we describe a simple user interface that allows a bidder to represent the constraints mentioned above. In the case of internet auctions, the user interface may be implemented as web pages, or as a function running on the bidder's internet access device. This invention includes the use of other interfaces that provide equivalent capability. The bids placed by a bidder are presented to the bidder in a table with a line for each bid. The bid line includes information that identifies the item, the bid amount, and may optionally include information on the status of the bid and on bidder specified constraints involving the bid.

The user interface allows the user to indicate the type of constraint he wishes to enter. The options include: budget constraint, maximum quantity constraint, minimum quantity constraint, precedence constraint, and general linear constraint. This indication may be made by a variety of means, including selecting a URI or clicking on a location in the web page, selecting a radio button labeled with the constraint type, or entering a characters in an editable field and submitting a request to the auction. Once the type of constraint has been selected, the bidder provides additional information.

- A budget constraint is specified by selecting a set of bids and indicating a budget value for that set. More than one constraint budget constraint can be specified in this fashion.

- A maximum quantity constraint is specified by selecting the set of items to be included in the constraint and indicating a maximum quantity for that set.

- A minimum quantity constraint is specified by selecting the set of items to be included in the constraint and indicating a minimum quantity for that set.

- A simple precedence constraint is specified by first specifying the item that has precedence, and then specifying the elements of a precedence set. If there are alternate precedence sets, the bidder specifies each alternate set separately.

- General linear constraints are specified by allowing the bidder to select a set of items, and then to specify a coefficient for each selected item and an upper and lower bound for each constraint.

Once all of the bidder constraints have been entered, an integer program for selecting the winning bids may be formulated. Let I denote the set of items, P denote the set of bidders, B_p denote the set of bids placed by bidder p, and B denote the entire set of bids. For convenience, we use B^i denote the set of bids for item i. For each $i \in I$ and each $p \in P$, let $v_{i,p} \geq 0$ denote the value of the bid bidder p has placed for item i. This value is 0 if there is no such bid. To designate whether the bid by bidder p for item i is included in the winning combination of bids, we use decision variables $x_{i,p}$ each of which must take either the value 0 (indicating that the bid i is not in the winning combination) or the value 1 (indicating that the bid i is in the winning combination). Note that we only include these variables for bidder-item combinations for which bids have been placed. Each of the user specified constraints described above can be modeled as a linear constraint in these binary variables.

Budget constraints are represented as follows: let $S_{l,p}^b$ be the set of items in the l-th budget constraint specified by bidder p and let $b_{l,p}$ be the budget specified for bidder p for this set of items. Then the budget constraint is given by

$$\sum_{i \in S_{l,p}^b} v_{i,p} x_{i,p} \leq b_{l,p}$$

5   Maximum cardinality constraints are represented as follows: let $S_{l,p}^{Max}$ be the set of items in the l-th maximum cardinality constraint specified by bidder p and let $q_{l,p}^{Max}$ be the maximum quantity specified for bidder p for this set of items. Then the maximum quantity constraint is given by

$$\sum_{i \in S_{l,p}^{Max}} x_{i,p} \leq q_{l,p}^{Max}$$

Minimum cardinality constraints are represented as follows: let $S_{l,p}^{Min}$ be the set of items in the l-th minimum cardinality constraint specified by bidder p and let $q_{l,p}^{Min}$ be the minimum quantity specified for bidder p for this set of items. We introduce an auxiliary variable, $z_{l,p}^{Min}$ which takes the value 1 if any of the items in $S_{l,p}^{Min}$ are selected, and 0 otherwise.

Then the minimum quantity constraint is represented by the following constraint.

$$z_{l,p}^{Min} \geq \sum_{i \in S_{l,p}^{Min}} x_{i,p} \geq q_{l,p}^{Min} z_{l,p}^{Min}$$

15   Note that if $z_{l,p}^{Min}$ is 0, then all of the $x_{i,p}$ variables corresponding to bids by bidder p for items in $S_{l,p}^{Min}$ must also be 0. If any of these variables takes the value 1, then at least $q_{l,p}^{Min}$ of them must take the value 1.

Simple precedence constraints are represented as follows: Let $S_{p,i}^{P0}$ denote the (single) precedence set for item i by bidder p. Then the simple precedence constraint is given by

$$\left| S_{p,i}^{P0} \right| x_{i,p} \leq \sum_{ji \in S^{P0}p,i} x_{j,p}$$

Representing multiple precedence constraints requires the addition of auxiliary 0-1 variables, used to indicate which of the precedence sets are satisfied. Let $S_{p,i,l}^{P}$ denote the l-th precedence set for item i by bidder p. Let $z_{p,i,l}^{P}$ be a 0-1 variable which takes the value 1 if and only bidder i is awarded all of the items in $S_{p,i,l}^{P}$. Then the alternate precedence constraint is represented by the following constraints.

$$\left| S_{p,i,l}^{P} \right| z_{p,i,l}^{P} \leq \sum_{j \in S_{p,i,l}^{P}} x_{j,p} \quad \text{for all } l$$

$$x_{i,p} \leq \sum_{l} z_{p,i,l}^{P}$$

General linear constraints are easy to represent. Let $S_{p,l}^{G}$ be the set of bids in the l-th general linear constraint specified by bidder p. Let $c_{p,l}^{U}$ and $c_{p,l}^{L}$ denote the bidder specified upper bound and lower bound respectively. For each item $j \in S_{p,l}^{G}$, let $a_{p,l,j}$ be the coefficient assigned by the bidder to item j. Then the general linear constraint is given by

$$c_{p,l}^{L} \leq \sum_{j \in S_{p,l}^{G}} a_{p,l,j} x_{p,j} \leq c_{p,l}^{U}$$

The set of winning bids can be determined by solving the following integer program:

$$Max \sum_{i,p} v_{i,p} x_{i,p}$$

Subject to $\quad \sum_p x_{i,p} \le 1$ $\qquad\qquad$ for all $i \in I$ $\qquad$ (1)

$$\sum_{i \in S_{l,p}^b} v_{i,p} x_{i,p} \le b_{l,p} \qquad\qquad \text{for all } p \in P, \text{ all } l \qquad (2)$$

$$\sum_{i \in S_{l,p}^{Max}} x_{i,p} \le q_{l,p}^{Max} \qquad\qquad \text{for all } p \in P, \text{ all } l \qquad (3)$$

$$z_{l,p}^{Min} \ge \sum_{i \in S_{l,p}^{Min}} x_{i,p} \ge q_{l,p}^{Min} z_{l,p}^{Min} \qquad\qquad \text{for all } p \in P, \text{ all } l \qquad (4)$$

$$\left| S_{p,i}^{P0} \right| x_{i,p} \le \sum_{j \in S_{p,i}^{P0}} x_{j,p} \qquad\qquad \text{for all } p \in P, \text{ all } l \qquad (5)$$

$$\left| S_{p,i,l}^{P} \right| z_{p,i,l}^{P} \le \sum_{j \in S_{p,i,l}^{P}} x_{j,p} \qquad\qquad \text{for all } p \in P, \text{ all } l, \text{all } i \qquad (6)$$

$$x_{i,p} \le \sum_l z_{p,i,l}^{P} \qquad\qquad \text{for all } p \in P, \text{ all } l, \text{ all } i \qquad (7)$$

$$c_{p,l}^{L} \le \sum_{j \in S_{p,l}^{G}} a_{p,l,j} x_{p,j} \le c_{p,l}^{U} \qquad\qquad \text{for all } p \in P, \text{ all } l \qquad (8)$$

$$x_{i,p} \in \{0,1\} \qquad\qquad \text{for all } i \in I, p \in P \qquad (9)$$
$$z_{p,i,l}^{P} \in \{0,1\} \qquad\qquad \text{for all} \in I, p \in P, \text{ all } l \qquad (10)$$
$$z_{p,,l}^{Min} \in \{0,1\} \qquad\qquad \text{for all} \in I, p \in P, \text{ all} l \qquad (11)$$

Constraint (1) says that each item is in at most one winning bid. If it is required that each item be

in exactly one winning bid (that is, that all of the items be sold) then the inequality should be

changed to an equality. In this case, a dummy bid should be added for each individual item should

be added, so that a feasible solution is guaranteed to exist. The remaining constraints have already

been explained.

Commercial integer programming solvers (such as the OSL product from IBM or the ILOG

product from CPLEX) can be used to solve problems of this form. However, the computation

time and the amount of computer memory required to solve large auctions may be excessive,

especially when there are a large number of user specific constraints. An alternative formulation, based on column generation techniques, may be more computationally attractive. Note that the bidder specific bids apply only to a single bidder (that is, each involves only the variables for a single value of the subscript p) and that the variables for different bidders are linked only by the item availability constraints.

Rather than consider each bid individually, and use constraints to indicate which combinations of bids can be selected simultaneously, we consider, for each bidder, the combinations of that bidder's bids that can be simultaneously in a winning solution. We call such combinations proposals. Essentially, a proposal is a set of bids from a single bidder, that satisfy all of the constraints specified by that bidder. If a bidder has placed a small number of bids, proposals from that bidder can be determined by enumeration. Otherwise, attractive (in terms of revenue) proposals can be generated for each bidder by solving a relatively small integer program involving only bids for that bidder.

For a bidder p we let Φ_p denote the collection of proposals for bidder p. Note that each proposal is a set of bids (or, equivalently, a bidder and a set of items). For a proposal $C_{k,p}$ in Φ_p, the amount that bidder p would pay for the proposal is denoted $w_{p,k}$ and is equal to the sum of the value of the bids in $C_{k,p}$, given by $w_{k,p} = \sum_{i \in C_{k,p}} v_{i,p}$.

Let $\Phi = \cup_p \Phi_p$ be the set of proposals. Then the winning combination of bids can be determined by selecting at most one proposal for each bidder, while ensuring that each item is in at most one selected proposal. For each $C_{k,p} \in \Phi_p$ we use 0-1 decision variables $y_{k,p}$ which indicates whether

the proposal is selected ($y_{k,p} = 1$) or not ($y_{k,p} = 0$). We include constraints that allow only one proposal to be selected from each bidder; however, since all bidder specific constraints are satisfied by all of the proposals, we do not need to include constraints to enforce this restriction. The set of winning bids can be determined by solving the following integer program:

$$Max \sum_{p} \sum_{C_{k,p} \in \Phi_p} w_{k,p} y_{k,p}$$

Subject to
$$\sum_{p,k: i \in C_{p,k}} y_{k,p} \leq 1 \qquad \text{for all } i \in I$$

$$\sum_{k} y_{k,p} \leq 1 \qquad \text{for all } p \in P$$

$$y_{k,p} \in \{0,1\} \qquad \text{for all } C_{k,p} \in \Phi$$

Although this formulation may potentially have far more decision variables than the original formulation, our computational experience indicates that for many integer programming problems, column generation based formulations may improve solvability. This faster computation time is due in part to the fact that all of the constraints in formulations like the one above have the same structure (sum of a set of variables is less than or equal to 1), and that the variables partition naturally in to sets of variables associated with each bidder. This partition, into so-called "special ordered sets" can be taken advantage of in commercial integer programming software.

It is important to note that the column generation approach works even when the columns of the matrix are not explicitly given, so long as the set of columns has a well defined structure (e.g., solutions to the set of constraints specified by the bidder) and new columns can be easily identified (e.g., by adjusting the objective function, and solving the small integer program representing a bidder's constraints). For details on the column generation technique, see Optimization Theory for Large Systems, Leon S. Lasdon, 1970, MacMillan Publishing, pp,146-148. For our formulation,

columns correspond to proposals, which have a well defined structure. Furthermore, the dual variables associated with the item constraints, which are typically computed in the process of solving the integer program or its linear programming relaxation can be easily used to adjunct the proposal values, and the dual variables associated with the bidder constraints can be used to

5   determine a threshold on the adjusted value of a proposal. Proposals are added to the restricted problem only if their adjusted value exceeds the threshold for the corresponding bidder.

Finally, we note that although the description above has been limited to the case of bidder specified constraints, analogous techniques can be used to represent seller constraints. One skilled in the art of mathematical optimization will readily observe how the above formulations

10   can be modified to reflect those constraints in the winner determination problem.

Detailed Description of a Preferred Embodiment of the Invention

Referring to the drawing, and more particularly to FIG. 1, there is shown a schematic diagram (10-28) of a computer implemented system for a combinatorial auction. One or more bidders participate in the auction. Two or more items are being auctioned. Each bidder uses a computer

15   interface to interact with the auction. Each bidder can enter bids, and each bidder is informed, through the interface of the status of his or her bids. The status of a bid is "SELECTED" if the bid is in the current set of winning bids. If no additional bids are entered, the "SELECTED" bids will become the "WINNING" bids. However, if additional bids are submitted, or if the value of an existing bid is increased, or if a bidder constraint is modified, a "SELECTED" bid may become

20   "UNSELECTED." Each bidder is allowed to edit his or her own bids although the types of edits may be significantly restricted by the auction rules. In general, we expect that bidders will be

allowed to increase the value of their bids. We also allow a bidder to specify and edit constraints on combinations of bids (or equivalently, combinations of items) that he is willing to be awarded. The specification and modification of constraints by bidders is a novel feature of this invention. In some auctions, bidders may also be allowed to withdraw bids, perhaps subject to some penalty.

5   We note that sellers may be allowed similar capability to place constraints and modify the set of items to be sold. This invention is not concerned with the particular bidding rules of the auction, only with the methods and systems used to select the winning bids.

Depending on the policy of the auction, as embodied in the authorization rules, bidders may be allowed to view all bids, or to view all winning bids. Viewing all bids is analogous to an open cry auction. The method and system described in this invention are applicable independent of the form of the authorization rules.

Bid data is maintained in a bid table, which is updated by a bid update function. The update function updates the bid table with information received from the bidders as well as with information received for the auction solver.

15   The Auction controller determines when the auction solver needs to run by monitoring the data in the bid table, or by receiving messages from the bid update function. The Auction controller also determines whether the solver is idle. In this figure we show only a single solver. For large auctions, with a large number of bidders, or for auction for very high value items, it may be desirable to run multiple solvers. In the first case, the solver run time may be greater than the

20   interval between bid activity, and running multiple solvers can give bidders faster feedback on the

status of their bids. In the second case, multiple solvers can be used to increase the confidence that optimal solutions are being obtained, so that the sellers' revenue is truly maximized.

FIG. 2 illustrates an implementation of the user interface, and the details of a single bidder's bid table. The details of the bid table are illustrated by example, although our methods can be applied

5   to other forms of bid data as long as the required elements (bid id, bidder id, item in the bid, value of the bid, and user specified constraints) are included. FIG. 2a (numeral 30) shows bids and constraints for one bidder; FIG. 2b (numeral 32) shows bids and constraints for a second bidder.

FIG. 3 illustrates, by example, the integer programming formulation for an auction involving 7 items and the two bidders in FIG 2. FIG. 3a (numeral 34) shows the initial formulation; FIG. 3b

10   (numeral 36) shows the column based formulation.

FIG. 4 (numerals 38-46) illustrates a implementations of the details of the auction solver based on the first integer programming formulation. The representation of a combinatorial auction as an integer program is known. However the inclusion of participant-specified constraints in auctions, whether combinatorial or non-combinatorial, and the formulation of the winner determination

15   problem with participant-specific constraints as an integer programming problem is a novel aspect of this invention.

FIG. 5 (numerals 48-60) illustrates an implementation of the details of the auction solver based on the second, column generation based integer programming formulation. The interpretation of

bidder-specified constraints as a means of specifying combinatorial bids, and the use of a column generation formulation is also a novel aspect of this invention.

In the column-generation based solver the proposal generator generates valid proposals for each bidder and each type. The proposals are generated either by enumeration and evaluation, or by

5   solving an integer program which involves only those variables and constraints related to a single bidder. This invention is not restricted to either of these methods for generating proposals.

As depicted in FIG. 6 (numeral 62), we also disclose the step of providing, to the bidder through a user interface, the ability to review and edit or reject the proposals that are generated based on the constraints specified by the bidder.

10   The integer program solver can make use of commercial software. In our implementation, we use functions included in the IBM product "Optimization Solutions and Library" (OSL), along with some problem specific code to steer the optimization process. Other commercial software for solving integer programs is available, and this invention is not limited to a particular choice of software or to a specific sequence of function calls used to invoke the selected software.

15   The solution to the first formulation is a set of selected bids. The solution to the second formulation is a set of selected proposals, which is readily translated into a set of winning bids. In both implementations, the list of winning bids is communicated to the bid data update function, which updates the bid status in the bid data table. If a bid is selected, its selected_flag is set to 1. Otherwise, its selected_flag is set to 0.

It is understood that the system and method of the present invention can be implemented using a plurality of separate dedicated or programmable integrated or other electronic circuits or devices (e.g., hardwired electronic or logic circuits such as discrete element circuits, or programmable logic devices such as PLDs, PLAs, PALs, or the like). A suitably programmed general purpose

5   computer, e.g., a microprocessor, microcontroller or other processor device (CPU or MPU), either alone or in conjunction with one or more peripheral (e.g., integrated circuit) data and signal processing devices can be used to implement the invention. In general, any device or assembly of devices on which a finite state machine capable of implementing the flow charts shown in the

10   figures can be used to enable the invention.